Programming Project: heat and flow

Hans Kuerten

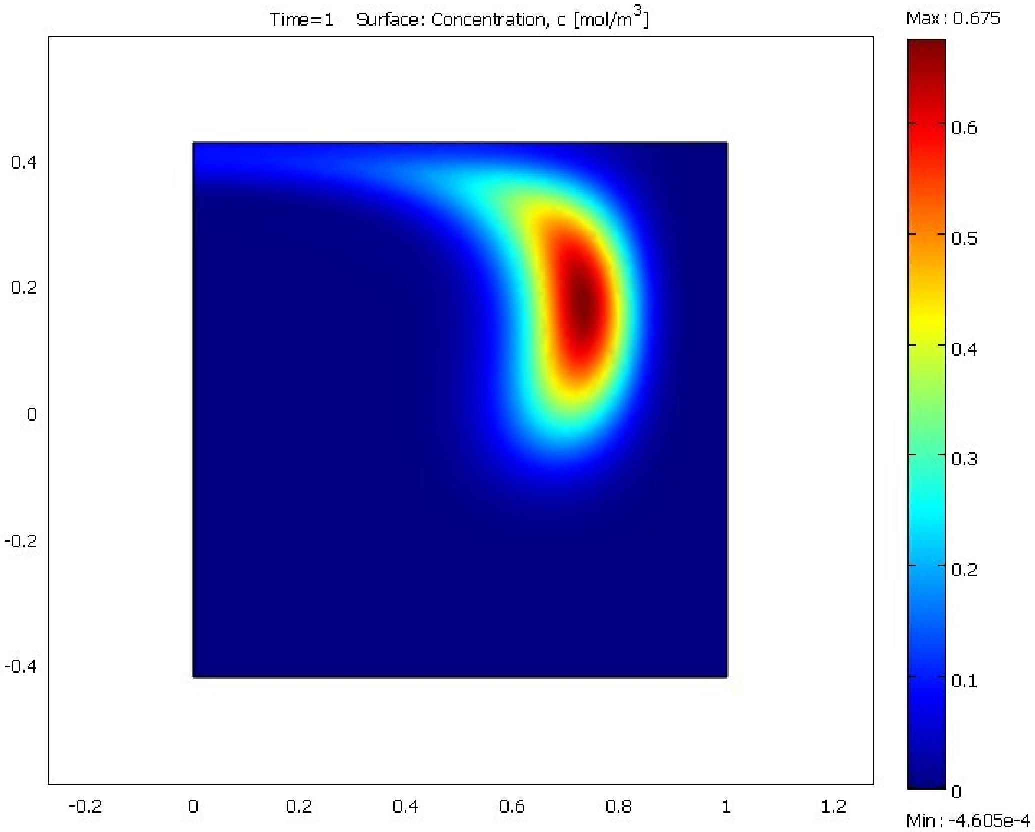
Several problems in heat and fluid flow are described by a convection-diffusion equation of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\mathbf{v}) = D\Delta u\,,$$

where **v** is the velocity field and *D* the diffusivity. Quantity *u* can denote for example temperature or concentration of a dissolved species. In this assignment a computer program will be developed in which this convection-diffusion problem is solved numerically. The emphasis is not on the numerical method, but on the development, testing and analysis of the computer code and on the interpretation of results. The numerical method will be rather simple and given. The development of the program will be done in several steps.

1. As a first problem a one-dimensional convection-diffusion equation will be taken of the form $\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = D\frac{\partial^2 u}{\partial x^2}$ with *v* and *D* constants. This problem will be solved with central differencing in space and a first-order explicit Euler forward method in time, i.e. if u_i^n denotes the solution in grid point x_i and at time level t^n, the numerical algorithm reads: $u_i^{n+١} = u_i^n - \frac{v\Delta t}{٢\Delta x}\left(u_{i+١}^n - u_{i-١}^n\right) + \frac{D\Delta t}{\Delta x^٢}\left(u_{i+١}^n - ٢u_i^n + u_{i-١}^n\right)$, where $\Delta\ t$ is the time step and $\Delta\ x$ the distance between two grid points. Since the right-hand side of the equation is completely known, the solution at the next time level can be calculated if boundary conditions and an initial condition are specified.
2. The next step is the programming of an implicit method for the same problem. This means that the two terms between brackets on the right-hand side of the discrete equation are not taken at the old, but at the new time level. The resulting set of equations can be written as a matrix-vector equation $Au^{n+1} = u^n$ with *A* a tri-diagonal matrix. First this system of equation will be solved with a program made by yourself. Next a library routine from the NAG library will be applied.
3. The second problem is a two-dimensional convection-diffusion equation with given velocity field, e.g. v_x=sin(x)sin(y), v_y=cos(x)cos(y), with a similar (implicit) numerical method as in the previous step. This leads to a matrix-vector equation with a sparse matrix for which several solution routines are available in the NAG

library.



4. As a final step the efficiency of the program will be studied, for example by varying the order of do-loops and the solution will be visualized. Attention will also be paid to efficient memory usage.

The program will be written in Fortran. This is available as Silverfrost FTN95. The NAG library used is version FL20.